UNIVERSIDAD DEL SINÚ
Elías Bechara Zainúm
Revista Científica Electronica
Facultad de Ciencias e Ingenierías
Ingenieria al Día

VARIACION DE PARAMETROS DE CRIPTOGRAFIA CON CURVAS ELIPTICAS USADOS EN LA FIRMA DIGITAL DE DATOS SOBRE UNA RED DE SENSORES INALAMBRICOS

(VARIATION OF PARAMETERS OF ELLIPTIC CURVE CRYPTOGRAPHY USED IN THE DIGITAL SIGNATURE OF DATA ON A WIRELESS SENSOR NETWOR)

Recibido: 31/03/2016 **Aprobado: 20/06/2016**

Javier Omar Contreras Rodriguez
Universidad Pontificia Bolivariana (UPB), Medellín, Colombia
Universidad de Los Andes, Mérida, Venezuela
javier.contreras@upb.edu.com, cjavierj@gmail.com

Reinaldo Nicolás Mayol Arnao
Universidad de Los Andes, Mérida, Venezuela
Universidad Pontificia Bolivariana (UPB), Medellín, Colombia
mayol@ula.ve, reinaldo.mayol@gmail.com

RESUMEN

En la actualidad, el auge de las aplicaciones de las redes de sensores inalámbricos (WSN = *Wireless Sensors Networks*) está generando una gran cantidad de información de carácter sensible que requiere un manejo confiable mediante la implementación de sistemas de seguridad de los datos compatibles con la naturaleza de estas redes. En ese sentido, cada vez más aumenta el interés por el uso de algoritmos de criptografía de clave pública con curvas elípticas (ECC = *Elliptic Curve Cryptography*) como una alternativa de menor consumo de recursos computacionales comparado con los algoritmos tradicionalmente usados, como por ejemplo, RSA (Rivest-Shamir-Adleman), Diffie-Hellman, otros. En este artículo, se revisa la construcción de un prototipo de sistema de seguridad usando ECC para la firma digital de datos (ECDSA = *Elliptic Curve Digital Signature Algorithm*) usando un control lógico basado en redes definidas por *software* (SDN = *Software Defined Networking*) para el control de funcionalidades básicas y que permita ajustar en tiempo real los parámetros del algoritmo ECDSA según el tipo de aplicación de la WSN.

Palabras Claves: Redes de sensores inalámbricos, Criptografía asimétrica, Curvas elípticas, Firma digital.

ABSTRACT

In the present times, the rise of Wireless Sensor Networks (WSN) applications is generating a lot of sensitive information that requires reliable operation by implementing

safety systems compliant with the nature of these networks. In this sense every day increases the interest in the use of algorithms of public key Elliptic Curve Cryptography (ECC) as an alternative to lower consumption of computing resources compared to traditionally used algorithms (eg. RSA, Diffie-Hellman, others). In this paper, it wil review the construction of a prototype system security using ECC (Elliptic Curve Cryptography) for the digital signature of data using a logic control based on SDN (Software Defined Netwroking) for the adjust on time real of the ECDSA (Elliptic Curve Digital Signature Algorithm) parameters according to the type of application of the WSN.

Key-words: *Wireless sensor network, Public key cryptography, Elliptic curves, Digital signature*.

INTRODUCCIÓN

En los últimos años, la seguridad de los datos se ha convertido en un campo de investigación de gran relevancia en el ámbito de las WSN, motivado al auge de estas redes para aplicaciones de monitoreo y control de parámetros de diversa índole en ambientes de carácter científico, médico, industrial y militar.

No obstante, proporcionar condiciones de confiabilidad durante el procesamiento y transmisión de datos sobre las WSN presenta aún un importante reto debido a sus características de limitados recursos de computación, ancho de banda y energía, largos periodos de operación ininterrumpidos, despliegue en zonas de difícil acceso, heterogeneidad, entre otras [1, 2].

En la actualidad, el desarrollo de protocolos y servicios de seguridad de los datos basados en ECC proveen niveles de seguridad similares a los ofrecidos por algoritmos criptográficos tradicionalmente usados, es decir, RSA, Diffie-Hellman, entre otros, pero con un menor consumo de recursos computacionales y de comunicación en las redes de datos. Es así como el uso de la ECC representa una opción para el despliegue de sistemas de seguridad de los datos acordes a la naturaleza de las WSN [3].

FIRMA DIGITAL CON CURVAS ELIPTICAS

El fundamento de la seguridad en la ECC descansa en la intratabilidad del problema de los logaritmos discretos en las curvas elípticas (ECDLP = *Elliptic Curve Discrete Logarithm Problem*). A nivel computacional, esto permite crear algoritmos criptográficos que utilizan claves de menor tamaño para alcanzar niveles de seguridad comparables a algoritmos criptográficos de clave pública tales como RSA. En la Tabla 1 se presenta una comparación del tamaño de clave necesaria para garantizar niveles similares de seguridad entre los algoritmos RSA y ECC [4].

El principal atractivo en el uso de claves de menor tamaño radica en el desarrollo de algoritmos criptográficos que consumen menor cantidad de recursos computacionales tales como ciclos de procesamiento y espacio de memoria del sistema donde sea implementado

dicho algoritmo. Esto incrementa de forma significativa la eficiencia en el uso de energía y ancho de banda del sistema, condiciones estas particularmente deseables en las WSN.

Tabla 1. Relación de tamaño de clave pública entre ECC y RSA

Tamaño de clave ECC (bits)	Tamaño de clave RSA (bits)	Relación de tamaño (bits)
160	1024	1:6
224	2048	1:9
256	3072	1:12
384	7680	1:20
521	15360	1:30

En el caso específico del ECDSA, el primer paso consiste en obtener los parámetros de dominio criptográfico y su interrelación en los procesos de generación del par de claves (privada y pública), generación y verificación de firma digital. A continuación, se resume cada uno de los procesos involucrados en el algoritmo ECDSA en función de los parámetros de dominio criptográfico [4].

- Generación del par de claves: Consiste en generar una clave privada *d* y una clave pública *Q* relacionadas matemáticamente a través de los parámetros de dominio criptográfico obtenidos. En un análisis básico, una vez seleccionada una curva elíptica *E(Zp)* y un punto base *P(xp, yp)* de orden *n*, se tiene que:
  - La clave privada será el valor de *d* correspondiente a número aleatorio en el intervalo acotado por [1, n -1].
  - La clave pública será $Q = dP$
- Generación de firma digital: Para firmar digitalmente un mensaje *m*, una entidad A requiere relacionar los dominios de parámetros criptográficos y el par de claves (*d, Q*) de la siguiente forma:
  - Por cada mensaje se genera un número *k* en el intervalo acotado por [1, n -1].
  - Se obtiene la relación $kP = (x1, y1)$
  - Se calcula $r = x1 \bmod n$, tomando *x1* como número entero. En caso de $r = 0$, se regresa al paso anterior.
  - Se calcula $(k^{-1}) \bmod n$
  - Se calcula $s = k^{-1}[H(m) + d.r] \bmod n$, donde *H(m)* corresponde al *hash* del mensaje (*m*) a firmar, calculado con algoritmo como SHA-1 (*Secure Hash Algorithm*).
  - Como resultado, los *r* y *s* corresponden a la firma del mensaje *m*.
- Verificación de firma digital: En el proceso de verificación del mensaje firmado, una entidad B obtiene una copia de los parámetros de dominio criptográfico definidos en la entidad A y los asocia con la clave pública *Q*. Así, de esa manera:
  - Se comprueba que los valores de *r* y *s* están en el intervalo acotado [1, n -1].

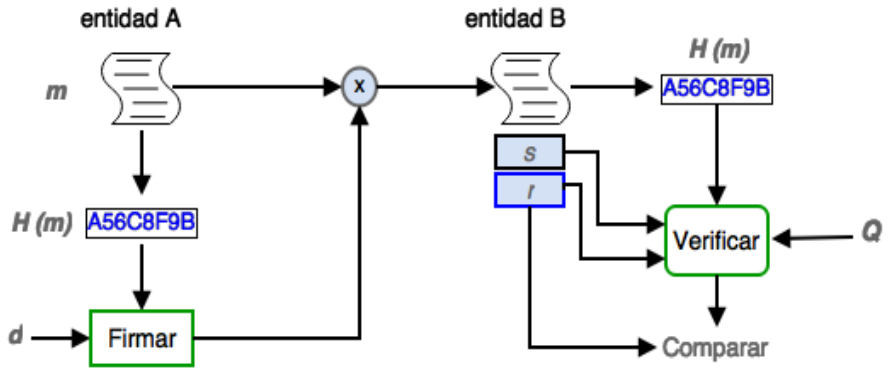
Revista Ingeniería al Día. ISSN: 2389 - 7309. Volumen 2 Edición No 2. Julio – Diciembre de 2016

Revista Ingeniería al Día. ISSN: 2389 - 7309. Volumen 2 Edición No 2. Julio – Diciembre de 2016

- Se calcula $w = s^{-1} \, mod \, n$
- Se calcula $u1 = (m).w \, mod \, n$
- Se calcula $u2 = r.w \, mod \, n$
- Se calcula $u1P + u2Q = (x0, y0)$
- Se calcula $v = x0 \, mod \, n$
- Si se cumple la relación $v = r$, se verifica la firma digital.

En la Figura 1, se esquematizan los procesos de generación y verificación de firma digital de datos involucrados en el algoritmo ECDSA [4].

Figura 1. Procesos de generación y verificación de firma digital de datos en ECDSA



SISTEMA DE SEGURIDAD USANDO ECDSA

Un prototipo de este sistema de seguridad consistió en implementar el ECDSA sobre una WSN desplegada sobre una arquitectura SDN. Con este fin, se recreó la comunicación entre un nodo de agregación y un nodo propagador interconectados en una red de sensores inalámbricos de arquitectura jerárquica basada en tecnología WiFi (IEEE 802.11g).

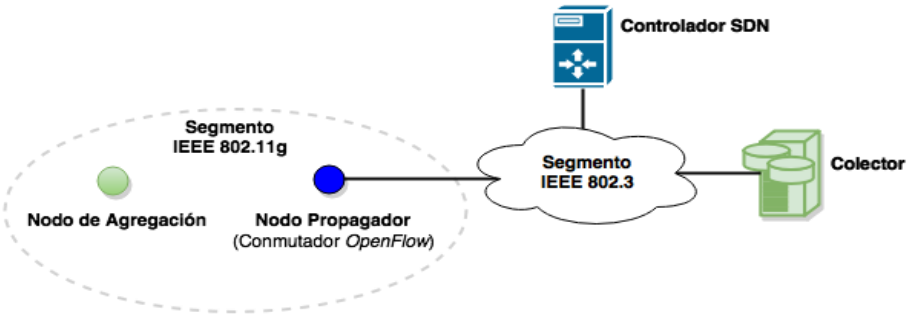
Por otra parte, el nodo propagador se conectó a un enlace de datos Ethernet (IEEE 802.3) emulando la conexión a una red de servidores en un centro de datos donde se instalaron un servidor colector y un controlador SDN. En la Figura 2 se muestra el diagrama general de red del prototipo construido incluyendo la asignación de direcciones IP por cada interfaz.

El algoritmo ECDSA empleado se codificó usando Python versión 3 como lenguaje de programación. Entre las ventajas consideradas para el uso de este lenguaje de programación se encuentra el uso de una sintaxis sencilla que permite generar códigos de fácil interpretación y completamente portables.

Python 3 también cuenta con soporte de bibliotecas propias y de acceso libre para la construcción de los códigos para el proceso de generación del par de claves (privada y pública), firma de datos y verificación de firmas en ECDSA. Estas bibliotecas son las *ecdsa-0.13* en las cuales se incluyen las definiciones de curvas elípticas según las

recomendaciones del NIST (*National Institute of Standards and Technology*).

Figura 2. Diagrama general prototipo propuesto (incluye dirección IP por interfaz)



Dispositivo	*Wi-Fi* (IEEE 802.11g)	*Ethernet* (IEEE 802.3)	*vsw0* (*OpenFlow*)
Nodo de Agregación	**10.1.1.1 /24**	***OFF***	NA
Nodo Propagador (Conmutador *OpenFlow*)	***vsw0*** *(bridge)*	***vsw0*** *(bridge)*	**172.16.255.2 /24**
Controlador SDN	***OFF***	**172.16.255.1 /24**	NA
Colector	***OFF***	**1.1.1.1 /24**	NA

NA = No Aplica

En el nodo de agregación se ejecutó un código para la generación del par de claves (privada y pública) para cada definición de curva elíptica. Seguidamente, las claves privadas se almacenaron en el nodo de agregación, mientras que las claves públicas se movieron hacia el servidor colector donde se verifica la integridad y autenticidad de los datos recibidos. En la Figura 3 se observa parte del código *generadorECDSA.py*.

Figura 3. Sección del código *generador ECDSA.py*

```
from ecdsa import SigningKey, NIST192p, NIST224p, NIST256p, NIST384p,
NIST521p

curve = NIST192p
 sk = SigningKey.generate(curve=curve)
```

La firma de datos se realizó en el nodo de agregación utilizando la clave privada correspondiente a la definición de curva elíptica seleccionada. Para los datos a ser firmados se usó un mensaje sintético (46 *bytes*) compuesto por un texto tal como “Data Sensor” sumado a un valor numérico generado por un contador de mensajes y a la salida de la función *time* de Python3 con la hora y fecha del sistema operativo. La Figura 4 muestra una sección del código *agregadorECDSA.py*.

Revista Ingeniería al Día. ISSN: 2389 - 7309. Volumen 2 Edición No 2. Julio – Diciembre de 2016

Figura 4. Sección del código *agregadorECDSA.py*

```
def Data_Sensor_Envio(nombre,conteo,sk):
mensaje="Data Sensor ("+ str(conteo) +"): "+ time.ctime(time.time() )
  firma = sk.sign(mensaje.encode('ascii'))
```

La verificación de la firma se realizó en el servidor colector a través de un código que, en primer lugar, valida la información de la etiqueta única que identifica al nodo de agregación y confirma la existencia de la clave pública que corresponde con la clave privada utilizada en la firma. Una vez que se confirmó la información anterior, se inició el proceso de verificación de firma que forma parte del algoritmo ECDSA. En la Figura 5 se observa parte de la secuencia que corresponde a la verificación de la firma de datos del código *colectorECDSA.py*.

Figura 5. Sección del código *colectorECDSA.py.*

```
def Verificar_Firma(vk,signature,mensaje):
    try:
      vk.verify(signature, mensaje)
      print ("good signature")
      except BadSignatureError:
      print ("bad signature ")
```

Una de las características de mayor relevancia del sistema de seguridad de los datos propuesto es la capacidad de programación a alto nivel y en tiempo real de sus funcionalidades básicas. En este sentido, en el controlador SDN se programó un código que permitió seleccionar en cualquier instante una nueva definición de curva elíptica empleada para la firma de datos. En la Figura 6 se observa el funcionamiento básico del control de definición de curvas elípticas.

En general, este control de funcionalidades básicas integraría a una red de sensores inalámbricos la capacidad de administrar los recursos computacionales de forma simultánea en todos los nodos sensores a través del ajuste en tiempo real de los parámetros del algoritmo ECDSA según se presenten variaciones en la configuración inicial de la red

Revista Ingeniería al Día. ISSN: 2389 - 7309. Volumen 2 Edición No 2. Julio – Diciembre de 2016

tales como cantidad de nodos, tipo de aplicaciones, cantidad y tipos de datos, entre otros. La Figura 7 muestra una sección con la tabla de selección en el código *controlECDSA.py*.

Figura 6. Funcionamiento control de definición de curvas elípticas

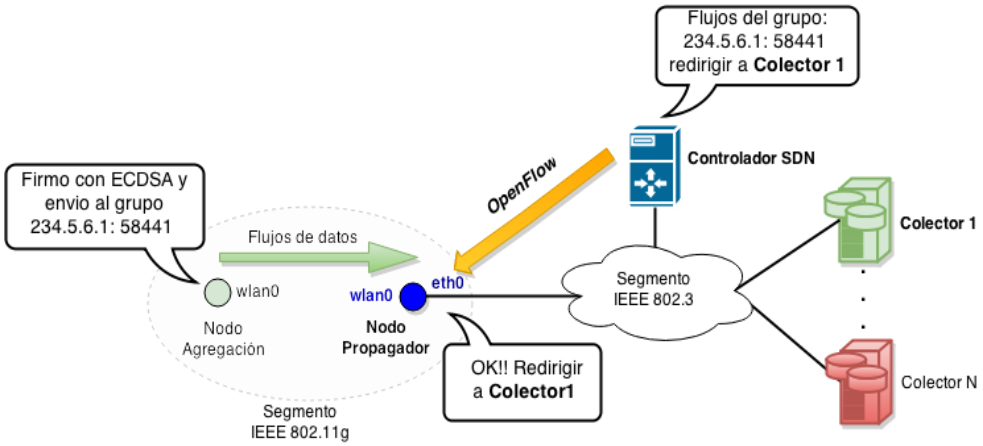


Figura 7. Sección del código *controlECDSA.py*

```
def UDP_Sender(threadName,host, port):
    conteo=0 #cuenta la cantidad de muestras generadas
    sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
    while True:
        print('Curvas Disponibles:')
        print('NIST192p    ...    ...    ...    0')
        print('NIST224p    ...    ...    ...    1')
        print('NIST256p    ...    ...    ...    2')
        print('NIST384p    ...    ...    ...    3')
        print('NIST521p    ...    ...    ...    4')
        try:
            x = int(input('Digite curva con la que desea trabajar:'))
        except ValueError:
            print ('Valor Invalido. Curva por defecto NIST192p')
```

Revista Ingeniería al Día. ISSN: 2389 - 7309. Volumen 2 Edición No 2. Julio – Diciembre de 2016

PRUEBAS DEL SISTEMA DE SEGURIDAD USANDO ECDSA

Para evaluar el pleno funcionamiento del prototipo construido, se iniciaron de forma secuencial cada uno de los componentes del prototipo, mientras se monitorearon en el nodo de agregación y el servidor colector de datos, los siguientes recursos computacionales:

1. Carga de procesamiento (%CPU)
2. Uso de memoria (MB)
3. Trafico en interfaz de red (bps)

Una vez iniciado el algoritmo ECDSA, se ejecutó el código *controlECDSA.py* desde el controlador SDN con el propósito de variar la definición NIST utilizada. Para cada definición y durante un periodo de tiempo de 30 minutos se obtuvieron gráficas de comportamiento para cada uno de los recursos computacionales antes mencionados, mientras se empleó una herramienta de *software* conocida como *Network Management Information System* (NMIS) basada en el protocolo de administración simple de red (SNMP = *Simple Networks Management Protocol*) y el protocolo de mensajes de control de Internet (ICMP = *Internet Control Message Protocol*).

El comportamiento de la carga de procesamiento (%CPU) observado evidenció la dependencia directa de este parámetro de la longitud de la clave utilizada para la firma digital con el algoritmo ECDSA. En las Figuras 8 y 9, se resume el comportamiento del %CPU en el nodo de agregación y el servidor colector durante las pruebas de funcionamiento pleno para diferentes definiciones del NIST.

Figura 8. Carga de procesamiento (%CPU) en nodo de agregación con prototipo en pleno funcionamiento

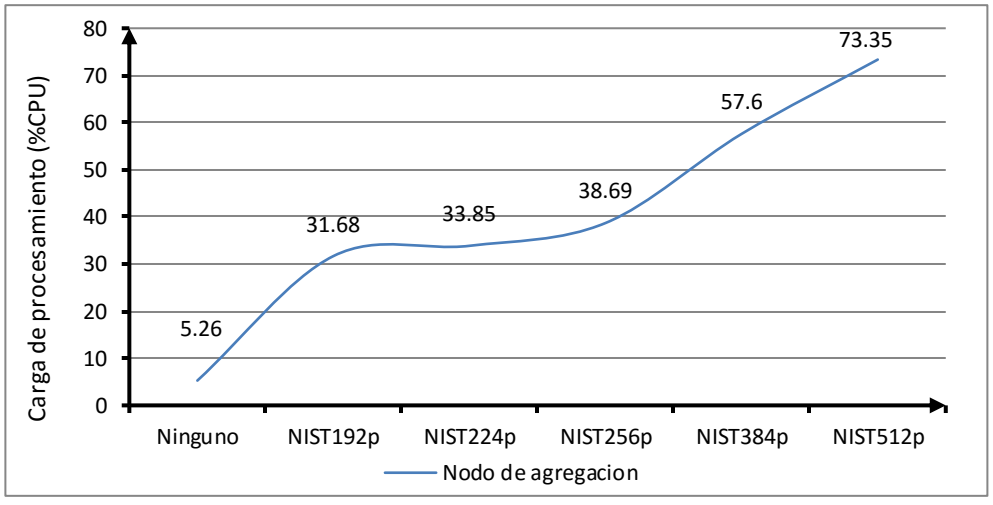
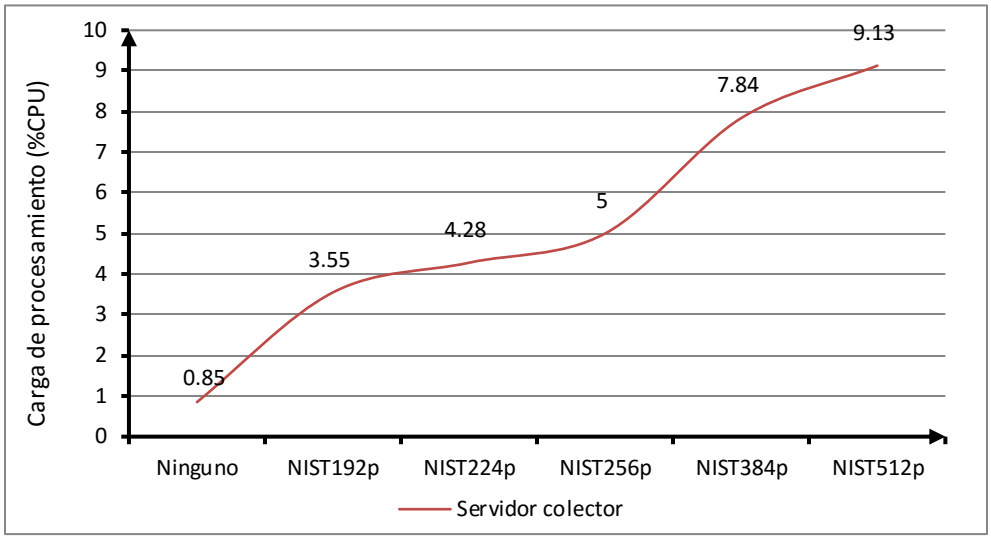


Figura 9. Carga de procesamiento (%CPU) en servidor colector con prototipo en pleno funcionamiento

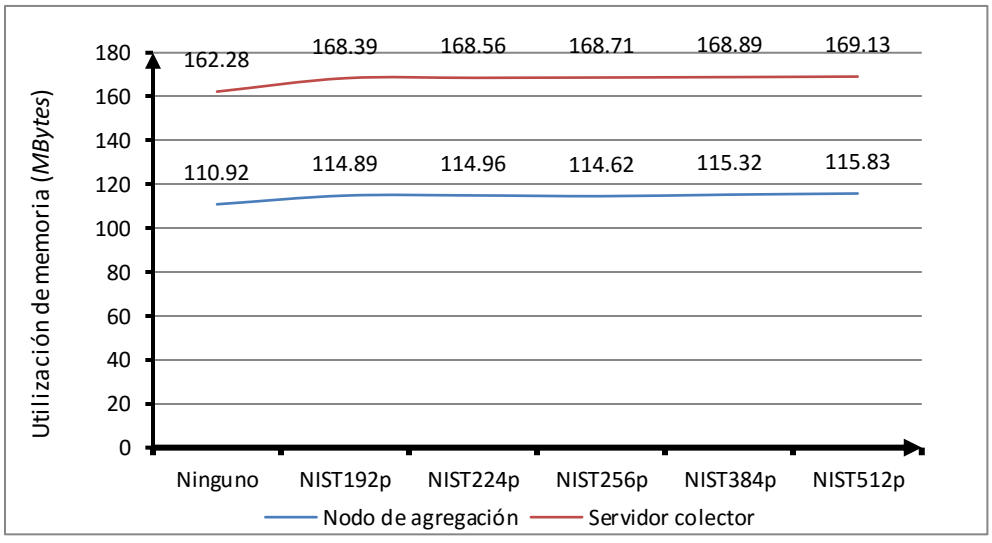
Revista Ingeniería al Día. ISSN: 2389 - 7309. Volumen 2 Edición No 2. Julio – Diciembre de 2016



En las Figuras 8 y 9 se observa como el comportamiento del %CPU no difiere de forma significativa entre las definiciones NIST192p, NIST224p y NIST256p en el nodo de agregación y el servidor colector. Un aumento relevante de este recurso se observó en las definiciones NIST384p y NIST521p.

En el caso del uso de memoria se observó un comportamiento constante de este recurso durante los procesos de firma digital de datos y verificación de firma con el algoritmo ECDSA durante todos los periodos en los que se variaron las definiciones del NIST. En la Figura 10 se detalla el comportamiento de este parámetro en el nodo de agregación y el servidor colector.

Figura 10. Utilización de memoria (MB) con prototipo en pleno funcionamiento



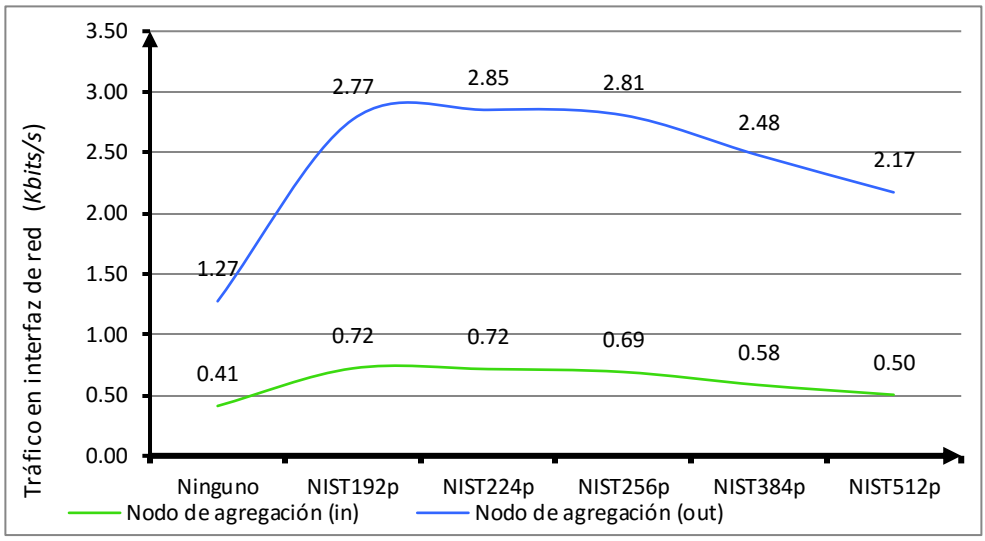
Aunque ambos dispositivos fueron aprovisionados con la misma capacidad de memoria de 512 MB, en la Figura 10 destaca la diferencia aproximada de 31,65 % en la utilización

Revista Ingeniería al Día. ISSN: 2389 - 7309. Volumen 2 Edición No 2. Julio – Diciembre de 2016

de memoria entre ambos dispositivos según mediciones realizadas antes de iniciar el algoritmo ECDSA. Se presume que esta diferencia está asociada a factores como la arquitectura computacional (ARM vs Intel Core) y la distribución de sistema operativo (Raspbian vs Debian 7) instaladas en cada dispositivo.

Por último, se monitoreó de manera simultánea el comportamiento del trafico entrante (*in*) y saliente (*out*) en las interfaces de red de los dispositivos nodo de agregación y servidor colector por un periodo de tiempo de un 30 minutos para cada defincion del NIST. En las Figuras 11 y 12, se muestra el parámetro tráfico en interfaz de red en función de las diferentes definiciones NIST empleadas.

Figura 11. Trafico en interfaz de red (kbps) en nodo de agregación con prototipo en pleno funcionamiento



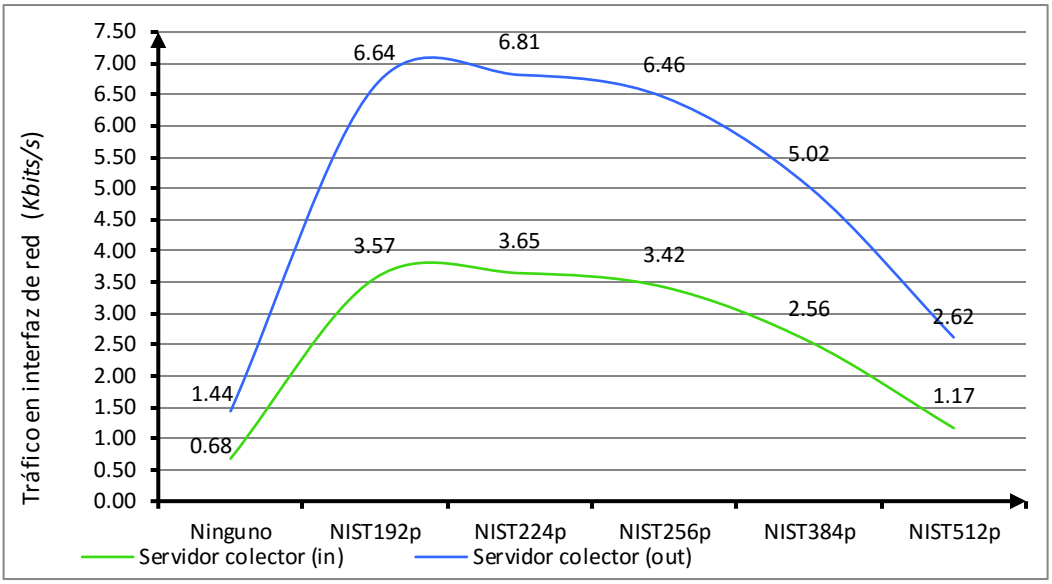
En el comportamiento observado en las Figuras 11 y 12, para el caso de las definiciones NIST192p, NIST224p y NIST256p se evidenció una zona de estabilidad en el comportamiento del tráfico en interfaz de red. Este resultado presenta una correlación directa con comportamiento del recurso %CPU discutido anteriormente en este mismo artículo.

CONCLUSIONES

De las pruebas exploratorias realizadas al prototipo desarrollado, se evidenció que la variación de parámetros de curvas elípticas entre las definiciones NIST192p, NIST224p y NIST256p está en el rango de funcionamiento estable del sistema propuesto para seguridad de los datos.

Figura 12. Trafico en interfaz de red (kbps) en servidor colector con prototipo en pleno funcionamiento

Revista Ingeniería al Día. ISSN: 2389 - 7309. Volumen 2 Edición No 2. Julio – Diciembre de 2016



Se observó también, en las pruebas realizadas, que los recursos de carga de procesamiento (%CPU) y tráfico en la interfaz de red (bits/s) presentaron mayor impacto durante los procesos involucrados en el algoritmo ECDSA en comparación con el recurso uso de memoria (bytes).

En términos generales, el sistema de seguridad propuesto permitió validar condiciones de integridad de los datos transmitidos sobre la WSN. De esta forma, fue posible diferenciar cuando la alteración de los datos se ha producido por un ataque generado por un tercero o es consecuencia de errores en la transmisión y procesamiento de la información.

Sin embargo, el prototipo construido demuestra debilidades en su capacidad de brindar autenticidad de los datos puesto que las claves públicas usadas en el colector de datos son cargadas de forma manual desde el nodo de agregación una vez se ha generado el par de claves (privada y pública). Esto trae como consecuencia, que el prototipo propuesto no sea escalable, condición de vital importancia en una red de sensores inalámbricos real.

Finalmente, en el sistema de seguridad propuesto se usaron dispositivos con suficientes recursos computacionales y de comunicación propiamente empleados en nodos de agregación. Aunque se prevé que las capacidades de estos dispositivos seguirán incrementándose, se considera necesario estudiar la implementación de sistemas ECDSA en nodos sensores con menores recursos y empleando protocolos de comunicación inalámbrica de bajo costo de energía (IEEE 802.15.4, Zigbee, 6LowPAN, otros).

REFERENCIAS BIBLIOGRÁFICAS

[1] Alcaraz C., Roman R. y López J. (2007). Análisis de Primitivas Criptográficas para Redes de Sensores. VI Jornadas de Ingeniería Telemática (JITEL'07), Málaga, España.

[2] Rehana J. (2009). Security of Wireless Sensor Network. Documento en línea. Disponible en http://cse.tkk.fi/en/públications/B/5/papers/Rehana_final.pdf (Consulta: enero 2016).

[3] Liu A. and Ning P. (2008). TinyECC: A Configurable Library for Elliptic Curve Cryptography in Wireless Sensor Networks. In 2008 International Conference on Information Processing in Sensor Networks (IPSN 2008). St. Louis , Missouri, USA

[4] Barker E., Barker W., Burr W., Polk W. and Smid M. (2012). Recommendation for Key Management – Part 1: General (Revision 3): Computer Security. NIST Special Públication 800-57, Revision 3 (Pp.1–147).

[5] Johnson D., Menezes A. and Vanstone S. (2004). The Elliptic Curve Digital Signature Algorithm Validation System (ECDSAVS). Documento en línea. Disponible en http://cs.ucsb.edu/~koc/ccs130h/notes/ecdsa-cert.pdf (Consulta: marzo 2016)

BIOGRAFIA DE LOS AUTORES

Javier Omar Contreras Rodríguez

Ingeniero Electricista de la Universidad de Los Andes (ULA), Venezuela, con más de 15 años de experiencia en el diseño y gestión de arquitecturas TIC (Tecnologías de Información y Comunicación) en organizaciones públicas y privadas del país. Participó en el despliegue y administración de la infraestructura de redes avanzadas de la Red Académica Venezolana (REACCIUN2) a través de la Corporación Parque Tecnológico de Mérida de la ULA (CPTM-ULA). Como instructor, ha colaborado con diferentes organizaciones internacionales tales como la Fundación Escuela Latinoamericana de Redes (Fundación EsLaRed), Red de Cooperación Latino Americana de Redes Avanzadas (RedCLARA), Registro de Recursos de Internet para Latinoamérica (LACNIC) y el International Centre for Theoretical Physics (ICTP). Actualmente, está radicado en Medellín, Colombia, donde se desempeña como especialista en Seguridad de la Información en la Caja de Compensación Familiar de Antioquia (COMFAMA), a la vez que participa como docente en la Universidad Pontificia Bolivariana (UPB) en el área de Redes TCP/IP y Seguridad de la Información. Cuenta con una Maestría en Telecomunicaciones en la Universidad de Los Andes (ULA), donde además funge como asesor del Grupo de Investigación en Telecomunicaciones (GITEL).

Reinaldo Nicolás Mayol Arnao

Ingeniero en Telecomunicaciones del Instituto Superior Politécnico “José Antonio Echeverría” de la ciudad de la Habana, Cuba. Tiene una Maestría en Informática de la Universidad de Los Andes (ULA), Mérida, Venezuela y grado de Especialista en Seguridad de la Información de Scientech de Venezuela. Desde 1993 se ha dedicado a la Administración de Redes y Servicios de Telecomunicaciones. En 1997 comenzó a trabajar en el Centro Nacional de Cálculo Científico de la Universidad de Los Andes y desde 1998 hasta 2002 dirigió las operaciones técnicas de la Red de Datos de dicha universidad. Fundó en el año 2000 el grupo de Seguridad de Cómputo de la Universidad de Los Andes. Entre 2002 y 2005 fue Gerente de Operaciones de Seguridad de la

Información de Scientech de Venezuela. Desde el año 2005 es profesor de la Universidad de Los Andes, Venezuela. Desde el año 2009 es profesor de Criptografía y Criptografía Aplicada en la Especialización de Seguridad Informática de la Universidad Pontificia Bolivariana (UPB), Colombia. Desde el año 2011 es el Coordinador Académico de la Especialización de Seguridad Informática de la Universidad Pontificia Bolivariana, Colombia. Desde 2000 se encarga de la coordinación del track de Seguridad Informática en la Escuela Latinoamericana de Redes y el Workshop para América Latina y el Caribe (WALC). Es candidato a Ph.D. en Ingeniería Electrónica en el áreas de Criptografía para redes de sensores. Desde 2014 es el Jefe de Seguridad de la Información (CISO) de la Caja de Compensación Familiar de Antíoquia, Colombia.

Revista Ingeniería al Día. ISSN: 2389 - 7309. Volumen 2 Edición No 2. Julio – Diciembre de 2016